1603 #13 11-18-02 F

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Tinku Acharya, et al.	§	Group Art Unit:	2625
Serial No.:	09/390,255	§	Examiner:	Timothy M. Johnson
Filed:	September 3, 1999	§		
For:	Wavelet Zerotree Coding of Ordered Bits	§	Atty. Dkt. No.:	ITL.0210US (P7057)

Board of Patent Appeals & Interferences
Commissioner for Patents
Washington, D.C. 20231

APPEAL BRIEF

Dear Sir:

Applicant hereby appeals from the Final Rejection dated July 26, 2002, finally rejecting claims 1-15.

I. REAL PARTY IN INTEREST

The real party in interest is Intel Corporation, the assignee of the present application by virtue of the assignment recorded at Reel/Frame 010390/0133.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

The application was originally filed with claims 1-15. These claims have been finally rejected under 35 U.S.C. § 102(e) and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

There are no unentered amendments.

V. SUMMARY OF THE INVENTION

Referring to Fig. 2, an embodiment 119 of a compression program in accordance with the invention may cause a processor 112 to encode wavelet coefficients in a bit-wise fashion. In this manner, instead of classifying the wavelet coefficients (as zerotree roots or isolated zeros, as examples), the processor 112 may produce codes to classify the bits of the wavelet coefficients. For example, in some embodiments, the processor 112 may classify a particular bit as being either a zerotree root, an isolated zero, a positive node or a negative node. Unlike conventional zerotree coding schemes, thresholds are not computed to identify insignificant values, as the "0" bit is treated as being insignificant and the "-1" and "1" bits are treated as being significant. Specification, pp. 4-5.

In this manner, the processor 112 may generate one of the following codes to classify a particular bit: a "P" code to indicate a positive node if the bit indicates a "1"; an "N" code to indicate a negative node if the bit indicates a "-1"; an "R" code to indicate that a "0" bit is a zerotree root ; and an "IZ" code to indicate that a "0" bit is an isolated zero. In some embodiments, a particular bit is classified as a negative node only if the bit is the most

significant nonzero bit and the bit indicates a "-1." For example, for a coefficient of "-3" that is represented by the three bits "-011," the processor 112 generates an N code to represent the middle bit. However, for this example, the processor 112 generates a P code to represent the least significant bit. Specification, p. 5.

For purposes of providing the wavelet coefficients, the processor 112 may, via wavelet transformations, decompose coefficients that represent pixel intensities of an original image. These wavelet coefficients, in turn, form subbands that are located in multiple decomposition levels. To classify the bits, the processor 112, in some embodiments, may execute the program 119 to process the bits based on their associated bit position, or order. In this manner, the bits of each bit order form a hierarchical tree that the processor 112 may traverse to classify each of the bits of the tree as being either a zerotree root, an isolated zero, a negative node or a positive node. Thus, as an example, the most significant bits of the wavelet coefficients(this bit may also be zero) are associated with one hierarchical tree (and one bit order), and the next most significant bits are associated with another hierarchical tree (and another bit order). Specification, p. 5.

For example, if the absolute maximum wavelet coefficient is represented by three bits (as an example), then all of the wavelet coefficients may be represented by three bits. Therefore, for this example, three hierarchical trees are formed. In this manner, the processor 112 produces a code for each bit based on its indicated value (i.e., "-1," "0," or "1") and possibly (if the bit indicates a "0") its position in the associated hierarchical tree. Specification, p. 5.

In some embodiments, the processor 112 indicates the P, N, IZ and R codes via a bit stream that progressively indicates a more refined (i.e., a higher resolution) version of the original image over time. For example, the processor 112 may use the bits "00" to indicate the "P" code, the bits "01" to indicate the "N" code, the bits "10" to indicate the "R" code and the bits "11" to indicate the IZ code. Other coding schemes are possible. The progressive nature of the bit stream is attributable to the order in which the processor 112 processes the bit orders. For example, in some embodiments, the processor 112 may process the bit orders in a most significant first fashion. Therefore, the processor 112 may initially produce code for all the bits that have the highest bit order, then produce code for all of the bits that have the next highest bit order, etc. As a result of this progressing coding, the resultant bit stream may initially indicate a coarser version of the original image. However, more refinements to the image are indicated by the bit stream over time, as the processor 112 produces the codes for the bits having the lower bit orders. Thus, in some embodiments, the resolution of the image that is indicated by the bit stream improves over time, a feature that may be desirable for bandwidth-limited systems. As a result, a decrease in resolution of the reconstructed image may be traded for a decrease in communication bandwidth. Specification, p. 6.

Referring to Fig. 3, in some embodiments, the processor 112 process the bits of each order in a predefined sequence. For example, for a particular bit order, the processor 112 may begin with the highest decomposition level and produce codes for the bits of the highest decomposition level before proceeding to produce codes for the bits of the next highest decomposition level. The processor 112 produces code(s) for the bit(s) of the LL subband and,

then for each decomposition level, produces code(s) for the bit(s) of the LH subband, subsequently, produces code(s) for the bit(s) of the HL subband and lastly, produces code(s) the bit(s) of the HH subband. Specification, p. 6.

As an example, the wavelet coefficients produced by a two level decomposition may be arranged in a matrix 40 that is depicted in Fig. 4. In this manner, the matrix 40 may be viewed as being subdivided into four quadrants 30a, 30b, 30c and 30d. The upper right 30b, lower left 30c and lower right 30d quadrants includes the coefficients for the LH, HL and HH subband images, respectively, of the first decomposition level. The coefficients for the LL, LH, HL and HH subband images of the second decomposition level are located in the upper left 32a, upper right 32b, lower left 32c and lower right 32d quadrants of the upper left quadrant 30a. The coefficients produced by further decomposition may be arranged in a similar manner. For example, for a third level of decomposition, the upper left quadrant 32a includes the wavelet coefficients of the LL, LH, HL and HH subbands of the third decomposition level. Specification, pp. 6-7.

If the coefficient matrix that indicates the pixel intensities for the original image is a 4X4 matrix, then the matrix 40 may be of the form that is depicted in Fig. 5. In this manner, the LL, LH, HL and HH subband images of the second decomposition level each have one coefficient, represented by "A" (for the LL subband image), "B" (for the LH subband image), "C" (for the HL subband image) and "D" (for the HH subband image), respectively. As depicted in Fig. 5, for the first decomposition level, the coefficients for the LH, HL and HH subband images are represented by the following respective matrices:

$$\begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix}, \begin{bmatrix} F_1 & F_2 \\ F_3 & F_4 \end{bmatrix}, \begin{bmatrix} G_1 & G_2 \\ G_3 & G_4 \end{bmatrix}$$

It is noted that each coefficient of the second decomposition level (except A), is associated with at least four coefficients of the first decomposition level, i.e., each coefficient of the first decomposition level has at least four descendant coefficients in the second decomposition level. Therefore, each bit in the first decomposition level has at least four descendent coefficients in the second decomposition level. Specification, p. 7.

For each bit order, the processor 112 may process the bits in the scanning sequence described above. If a particular bit indicates a "1" or a "-1," then the processor 112 generates the P or N code and proceeds to process the next bit in the scanning sequence. However, if a particular bit indicates a "0," then the processor 112 may trace the bit through its descendants to determine if the bit is an isolated zero or a zerotree root. The coefficients in the LL subband are simply entropy encoded. Specification, pp. 7-8.

As an example, to produce the code for the least significant bit (called D(1)) of the D coefficient (located in the HH subband of the second decomposition level), the processor 112 determines whether the D(1) bit indicates a "0." If so, the processor 112 evaluates the descendant bits $G_1(1)$, $G_2(1)$, $G_3(1)$ and $G_4(1)$ of the subband HH of the first decomposition level in search of a "1" or "-1," as indicated in Fig. 6. If one of these bits indicates a "1" or "-1," then the D(1) bit is an isolated zero. Otherwise the D(1) bit is a zerotree root. Specification, p. 8.

As a numeric example, a 4X4 coefficient matrix that indicates pixel intensities for an image may undergo a two level decomposition to form the following matrix:

$$\begin{bmatrix} 4 & 1 & 1 & 2 \\ -2 & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Because the maximum absolute value is "4," three bits may be used to represent the coefficients, as depicted in the following matrix:

$$\begin{bmatrix} 100 & 001 & 001 & 010 \\ -010 & 000 & 000 & 001 \\ 000 & 011 & 000 & 000 \\ 000 & 001 & 000 & 000 \end{bmatrix}$$

Therefore, the processor 112 begins the encoding by generating codes for the third order bits (i.e., the most significant bits, which may be zero also) of the coefficients. More particularly, to generate the codes for the third order bits, the processor 112 follows the path 28 (see Fig. 5) and produces the appropriate code for the third bit of each coefficient along the path 28. If a particular bit indicates a "0," then the processor 112 evaluates the descendants of the bit to find isolated zeros and zeroroots. The coding of the third order bits by the processor 112 produces the following codes (listed in the order of production): P,R,R,R. Subsequently, the processor 112 produces the codes for the second order bits (listed in order of production): IZ,IZ,N,R,IZ,P,IZ,IZ,IZ,P,IZ,IZ. Lastly, the processor 112 produces the codes for the first order bits (listed in order of production): IZ,P,IZ,R,P,IZ,IZ,P,IZ,P,IZ,P. As described above, the processor 112 may indicate the codes via a two bit coding scheme and transmit the codes as produced via a bit stream. Specification, pp. 8-9.

As an example, another processor 200 (see Fig. 2) may use the bit stream to reconstruct the coefficient matrix that indicates the pixel in intensities of the original image in the following manner. Before the decoding begins, the processor 200 first receives an indication from the processor 112 that three levels of coding (i.e., one level for each bit order) have been used. After obtaining this information, the processor 200 may reconstruct the original coefficient matrix using the codes in the order that the codes are produced. More particularly, the processor 200 may use the codes produced by the coding of the bits of the third bit order (i.e., the first level of coding) to produce the following matrix:

$$\begin{bmatrix} 100 & 000 & 000 & 000 \\ 000 & 000 & 000 & 000 \\ 000 & 000 & 000 & 000 \\ 000 & 000 & 000 & 000 \end{bmatrix}$$

The processor 200 may use this matrix to reconstruct a coarse version (i.e., a lower resolution version) of the original image. However, if a more refined version is desired, the processor 200 may use the codes that are produced by the coding of the second bit order (i.e., the second level of coding) to produce the following matrix:

$$\begin{bmatrix} 100 & 000 & 000 & 000 \\ -010 & 000 & 000 & 000 \\ 000 & 010 & 000 & 000 \\ 000 & 000 & 000 & 000 \end{bmatrix}$$

Finally, if the processor 200 uses the codes that are produced by the coding of the bits of the first order (i.e., the third level of coding), the processor 200 produces the original matrix of decomposed wavelet coefficients. Specification, pp. 9-10.

Referring to Fig. 7, to summarize, the compression program 119, when executed by the processor 112 may cause the processor 112 to perform the following procedure to produce the above-described coding. First, the processor 112 may express (block 72) a matrix of decomposed coefficients in a signed binary representation. Next, the processor 112 may determine (block 74) the number of digits that are needed to represent the absolute value of the maximum wavelet coefficient. This processor 112 uses a variable (called n) that indicates the current bit order being processed by the processor 112. In this manner, the processor 112 uses a software loop to process the bits, one bit order at a time. To accomplish this, the processor 112 produces codes (block 76) for the bits of the current bit order the using the techniques described above. Subsequently, the processor 112 determines (diamond 78) whether the rate of transmitted bits may exceed a predetermined bit rate. If so, the processor 112 terminates the coding for the current image to comply with the predetermined bit rate. Otherwise, the processor 112 determines (diamond 80) if all bit orders have been processed, i.e., the processor 112 determines if n equals "1." If not, the processor 112 decrements (block 75) the order that is indicated by the n variable by one and proceeds to block 76 to traverse the loop another time to produce codes for the bits of another bit order. Otherwise, the coding is complete. Specification, p. 10.

Referring back to Fig. 2, in some embodiments, the processor 112 may be part of a computer system 100. The computer system 100 may include a bridge, or memory hub 116, and the processor 112 and the memory hub 116 may be coupled to a host bus 114. The memory hub 116 may provide interfaces to couple the host bus 114, a memory bus 129 and an Accelerated Graphics Port (AGP) bus 111 together. The AGP is described in detail in the Accelerated

Graphics Port Interface Specification, Revision 1.0, published on July 31, 1996, by Intel Corporation of Santa Clara, California. A system memory 118 may be coupled to the memory bus 129 and store the compression program 119. As described above, the compression program 119, when executed by the processor 112, may cause the processor 112 to provide wavelet coefficients that indicate an image and represent each wavelet coefficient as a collection of ordered bits. The processor 112 codes the bits of each order to indicate zerotree roots that are associated with the order. Specification, pp. 10-11.

Among other features of the computer system 100, a display controller 113 (that controls the display 114) may be coupled to the AGP bus 11. A hub communication link 115 may couple the memory hub 116 to another bridge circuit, or input/output (I/O) hub 120. In some embodiments, the I/O hub 120 includes interfaces to an I/O expansion bus 125 and a Peripheral Component Interconnect (PCI) bus 121. The PCI Specification is available from The PCI Special Interest Group, Portland, Oregon 97214. Specification, p. 11.

A modem 140 may be coupled to the PCI bus 121 to a telephone line 142. In this manner, the modem 140 may provide an interface that permits the bit stream that is produced by the processor 112 to be communicated to the processor 200. The I/O hub 120 may also include interfaces to a hard disk drive 132 and a CD-ROM drive 133, as examples. An I/O controller 117 may be coupled to the I/O expansion bus 125 and receive input data from a keyboard 124 and a mouse 126, as examples. The I/O controller 117 may also control operations of a floppy disk drive 122. Copies of the program 119 may be stored on, as examples, the hard disk drive 132, a diskette or a CD-ROM, as just a few examples. Specification, p. 11.

In the context of this application, the phrase "computer system" may generally refer to a processor-based system and may include (but is not limited to) a graphics system, a desktop computer or a mobile computer (a laptop computer, for example), as just a few examples. The term "processor" may refer to, as examples, at least one microcontroller, X86 microprocessor, Advanced RISC Machine (ARM) microprocessor, or Pentium-based microprocessor. The examples given above are not intended to be limiting, but rather, other types of computer systems and other types of processors may be included in embodiments of the invention. Specification, pp. 11-12.

VI. ISSUES

A. Can a reference that does not teach representing wavelet coefficients as collections of ordered bits and coding the bits of each order to indicate zerotree roots that are associated with the order anticipate claims 1-6?

B. Can a reference that does not teach a storage medium that stores instructions to cause a processor to represent wavelet coefficients as collections of ordered bits and code the bits of each order to indicate zerotree roots that are associated with the order anticipate claims 7-11?

C. Can a reference that does not teach a memory that stores a program to cause a processor to represent wavelet coefficients as collections of ordered bits and code the bits of each order to indicate zerotree roots that are associated with the order anticipate claims 12-15?

VII. GROUPING OF THE CLAIMS

Claims 1-6 can be grouped together; claims 7-11 can be grouped together; and claims 12-15 can be grouped together.

VIII. ARGUMENT

All claims should be allowed over the cited references for the reasons set forth below.

A. Can a reference that does not teach representing wavelet coefficients as collections of ordered bits and coding the bits of each order to indicate zerotree roots that are associated with the order anticipate claims 1-6?

The method of claim 1 includes providing wavelet coefficients that indicate an image and representing each wavelet coefficient as a collection of ordered bits. The method includes coding the bits of each order to indicate zerotree roots that are associated with the order.

The Examiner rejects independent claim 1 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,222,941 B1 (herein called "Zandi"). Zandi generally teaches an apparatus for compression using reversible embedded wavelets.

More specifically, Zandi teaches zerotree encoding coefficients, not representing each wavelet coefficient as a collection of ordered bits and coding the bits of each order to indicate zerotree roots that are associated with the order. Thus, claim 1 is directed to a bitwise zerotree coding scheme, and in contrast, Zandi is directed to a coefficient-based zerotree coding scheme. The Examiner refers to the discussion (in Zandi) that begins on line 59 of column 22 and ends on line 25 of column 27. However, this discussion relates to zerotree encoding wavelet coefficients, not coding bits of each bit order to indicate zerotree roots that are associated with the order. Although beginning on line 27 of column 24 Zandi teaches determining whether a particular wavelet coefficient is a zerotree root based on a most significant bit S_A, Zandi neither teaches nor suggests coding the bits of each bit order. In this manner, Zandi teaches analyzing the most significant bit of a particular wavelet coefficient to determine whether the coefficient should be

classified in an A-group or a B-group. However, Zandi does not teach or even suggest coding the bits of each bit order to indicate zerotree roots that are associated with the order.

In the Final Office Action, the Examiner states "when zerotree coding, the very purpose is 'to indicate zerotree roots' that are associated with the wavelet coefficient ordered bits, as claimed, in order to encode the tree at a very low cost." Final Office Action, 4. The Examiner further adds in the Final Office Action, "to indicate a zerotree roots, each of the ordered bits of the wavelet coefficients are tested for significance." *Id.*

However, contrary to the Examiner's contentions, Zandi does not teach the claim limitations. In this manner, as set forth above, Zandi is directed to zerotree encoding *coefficients*, not coding bits to indicate zerotree roots that are associated with *bit orders*. Therefore, regardless of whether the Examiner's assertion that to indicate zerotree roots of each order, bits of wavelet coefficients are tested for significance is true or not, testing ordered bits to determine whether a particular wavelet coefficient is or is not a zerotree root does not teach or even suggest locating zerotrees for bit orders of wavelet coefficients. Furthermore, neither Zandi nor any of the other references cited by the Examiner support the Examiner's contention that the very purpose of zerotree coding is to indicate zerotree roots that are associated with wavelet coefficient ordered bits. Thus, Zandi does not teach or suggest coding the bits of each order to indicate zerotree roots that are associated with the order.

Therefore, the § 102(e) rejections of claims 1-6 are improper and should be reversed.

B. Can a reference that does not teach a storage medium that stores instructions to cause a processor to represent wavelet coefficients as collections of ordered bits and code the bits of each order to indicate zerotree roots that are associated with the order anticipate claims 7-11?

The article of claim 7 includes a storage medium that is readable by a processor-based system. This storage medium stores instructions to cause a processor to provide wavelet coefficients that indicate an image and represent each wavelet coefficient as a collection of ordered bits. The instructions also cause the processor to code the bits of each order to indicate zerotree roots that are associated with the order.

The Examiner rejects independent claim 7 under 35 U.S.C. § 102(e) as being unpatentable over Zandi. However, Zandi neither teaches nor suggests coding bits of each order to indicate zerotree roots that are associated with the order. In this manner, Zandi teaches coding wavelet coefficients, not bit orders. Furthermore, Zandi does not teach or suggest instructions to cause a processor to code bits of each order to indicate zerotree roots that are associated with the order.

Thus, the § 102(e) rejections of claims 7-11 are improper and should be reversed.

C. Can a reference that does not teach a memory that stores a program to cause a processor to represent wavelet coefficients as collections of ordered bits and code the bits of each order to indicate zerotree roots that are associated with the order anticipate claims 12-15?

The computer system of claim 12 includes a processor and a memory. The memory stores a program to cause the processor to provide wavelet coefficients that indicate an image and represent each wavelet coefficient as a collection of ordered bits. The program also causes

the processor to code the bits of each order to indicate zerotree roots that are associated with the order.

The Examiner rejects independent claim 12 under 35 U.S.C. § 102(e) as being unpatentable over Zandi. However, Zandi does not teach or suggest a program to cause a processor to code bits of each order to indicate zerotree roots that are associated with the order.

Thus, the § 102(e) rejections of claims 12-15 are improper and should be reversed.

IX. CONCLUSION

Applicant requests that each of the final rejections be reversed and that the claims subject to this appeal be allowed to issue.

Respectfully submitted,

Date: 10/23/02

Fred G. Pruner, Jr., Reg. No. 40,779
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024-1805
713/468-8880 [Phone]
713/468-8883 [Facsimile]

APPENDIX OF CLAIMS

The claims on appeal are:

1. A method comprising:

providing wavelet coefficients that indicate an image;

representing each wavelet coefficient as a collection of ordered bits; and

coding the bits of each order to indicate zerotree roots that are associated with the order.

2. The method of claim 1, wherein the act of coding the bits comprises:

determining which of the bits indicate zeros; and

classifying each zero as either an isolated zero or a zerotree root.

3. The method of claim 2, wherein some of the wavelet coefficients are descendants of some of the other wavelet coefficients, and wherein the act of determining comprises:

traversing a descendant tree from a bit associated with one of said some of the wavelet coefficients to bits associated with said other wavelet coefficients to locate the zerotree roots.

4. The method of claim 1, wherein the act of providing comprises:

producing different levels of the code, each level being associated with a different resolution of the image.

5. The method of claim 4, wherein the levels that are associated with lower resolution are associated with higher orders.

6. The method of claim 1, wherein the act of providing wavelet coefficients comprises:

providing intensity level coefficients that indicate pixel intensities of the image; and

transforming the intensity level coefficients into wavelet subbands.

7. An article comprising a storage medium readable by a processor-based system, the storage medium storing instructions to cause a processor to:

provide wavelet coefficients that indicate an image,

represent each wavelet coefficient as a collection of ordered bits, and

code the bits of each order to indicate zerotree roots that are associated with the order.

8. The article of claim 7, the storage medium comprising instructions to cause the processor to:

determine which of the bits indicate zeros, and

classify each zero as either an isolated zero or a zerotree root.

9. The article of claim 8, wherein some of the wavelet coefficients are descendants of some of the other wavelet coefficients, the storage medium comprising instruction to cause the processor to:

traverse a descendant tree from a bit associated with one of said some of the wavelet coefficients to bits associated with said other wavelet coefficients to locate the zerotree roots.

10. The article of claim 7, the storage medium comprising instructions to cause the processor to:

produce different levels of the code, each level being associated with a different resolution of the image.

11. The article of claim 10, wherein the levels that are associated with lower resolutions are associated with higher orders.

12. A computer system comprising:

a processor; and

a memory storing a program to cause the processor to:

provide wavelet coefficients that indicate an image,

represent each wavelet coefficient as a collection of ordered bits, and

code the bits of each order to indicate zerotree roots that are associated with the order.

13. The computer system of claim 12, wherein the program causes the processor to code the bits by determining which of the bits indicate zeros and classifying each zero as either an isolated zero or a zerotree root.

14. The computer system of claim 13, wherein some of the wavelet coefficients are descendants of some of the other wavelet coefficients, and wherein the processor determines which of the bits are zeros by traversing a descendant tree from a bit associated with one of said

some of the wavelet coefficients to bits associated with said other wavelet coefficients to locate the zerotree root.

15. The computer system of claim 12, wherein the program causes the processor to provide the wavelet coefficients by producing different levels of the code, each level being associated with a different resolution of the image.

TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. ITL.0210US

In Re Application Of: **Tinku Acharya, et al.**

Serial No.	Filing Date	Examiner	Group Art Unit
09/390,255	September 3, 1999	Timothy M. Johnson	2625

Invention: **Wavelet Zerotree Coding of Ordered Bits**

RECEIVED
OCT 30 2002
Technology Center 2600

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on **October 22, 2002**

The fee for filing this Appeal Brief is: **$320.00**

☒ A check in the amount of the fee is enclosed.

☐ The Commissioner has already been authorized to charge fees in this application to a Deposit Account. A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **20-1504**
A duplicate copy of this sheet is enclosed.

Signature

Dated: **October 23, 2002**

Fred G. Pruner, Jr., Reg. No. 40,779
Trop, Pruner & Hu, P.C.
8554 Katy Freeway, Suite 100
Houston, Texas 77024
(713) 468-8880 [Phone]
(713) 468-8883 [Fax]

21906
PATENT TRADEMARK OFFICE

cc:

I certify that this document and fee is being deposited on 10/23/2002 with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Signature of Person Mailing Correspondence

Debra Cutrona

Typed or Printed Name of Person Mailing Correspondence